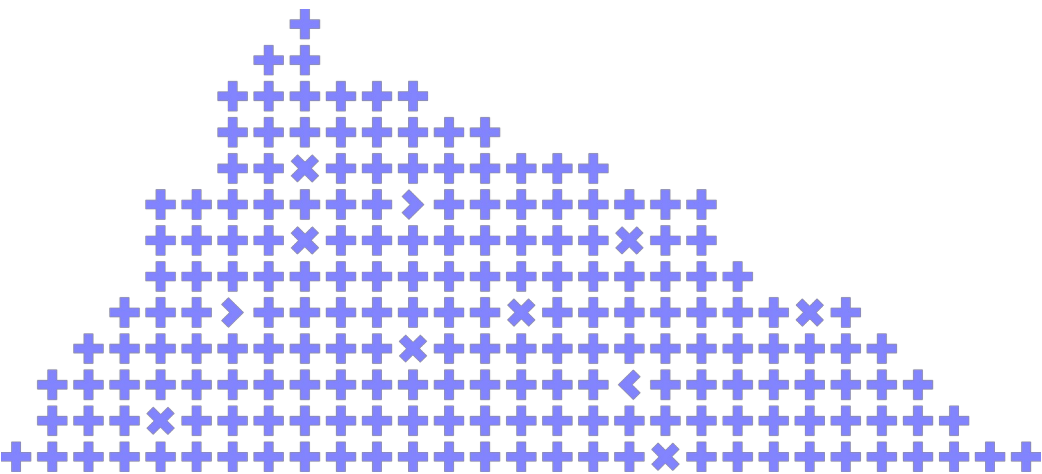


From MVP to Reality. Transition Problems and Solutions

Aleksei Dashkevich



Co-organizer

Yandex

ALEKSEI



Solution architect



Responsibilities

design the architecture for IT products



Digital transformation

in telecom companies using TM Forum practices



Also interested in

educational projects for IT specialists

“IT_simulator”



In love

with snowboarding and mountains

WE WILL TALK ABOUT



The product
automates store
operations

It affects the efficiency of
the employees and
customer's NPS

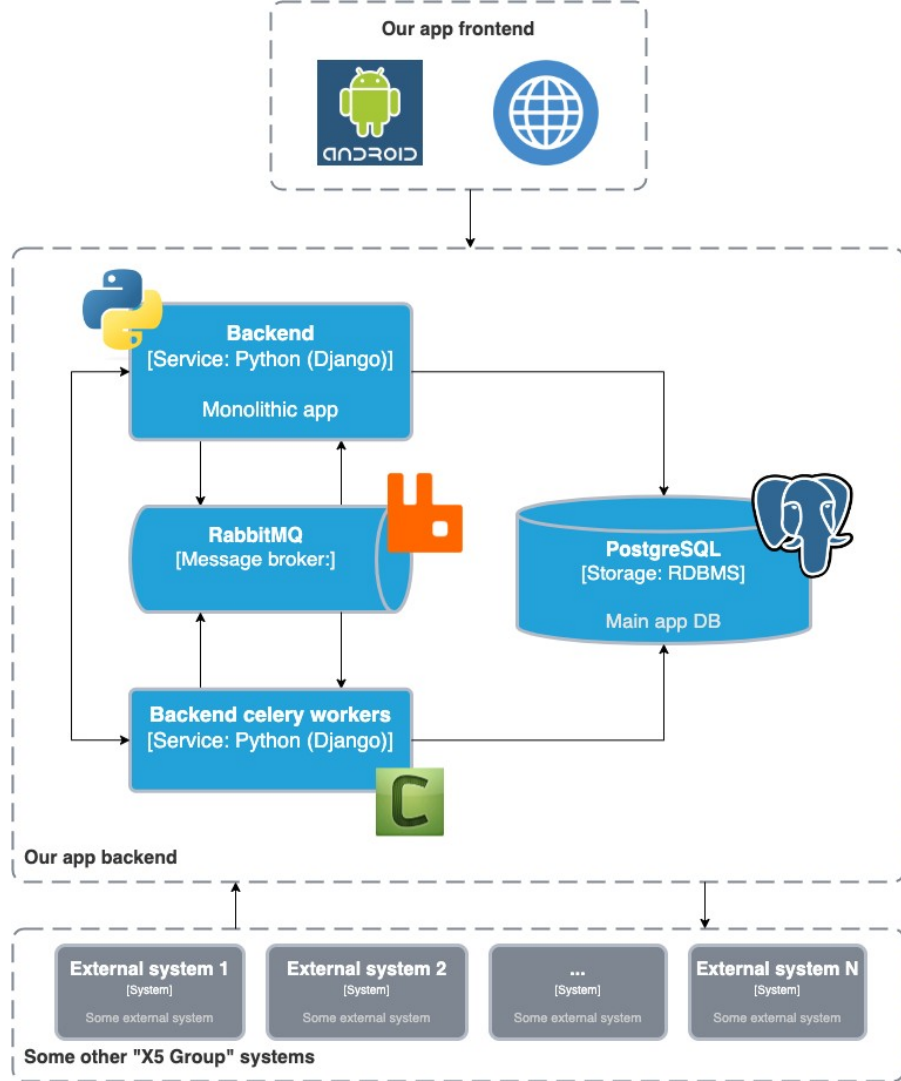


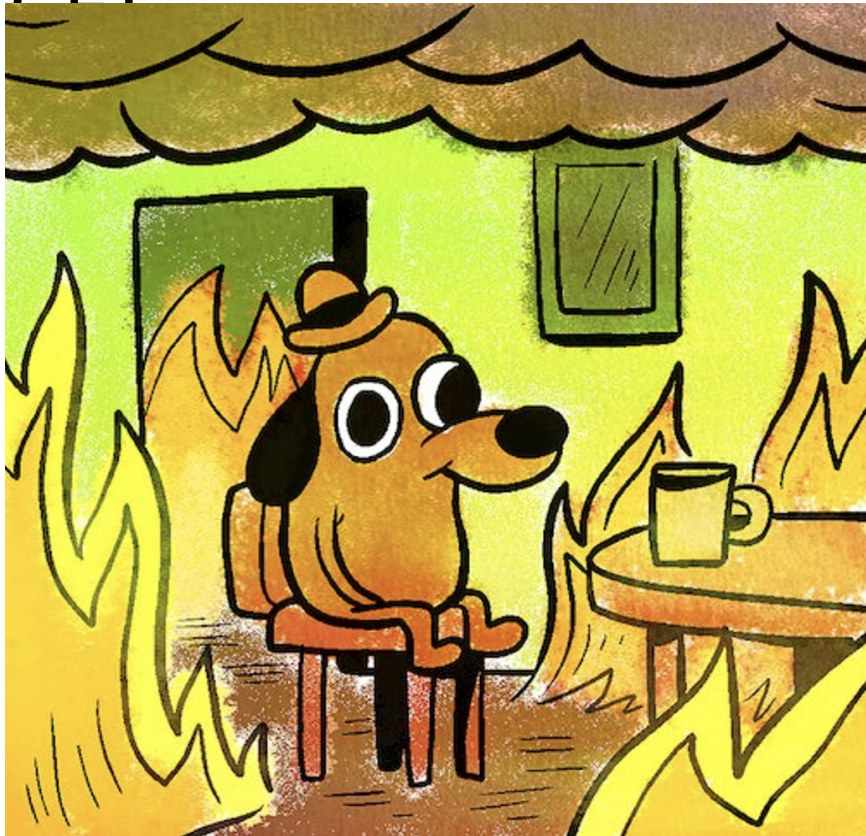
What problems did we
face?







while scaling from 20 to
~1000 stores in 2 months

ABOUT MVP ARCHITECTURE

- ✓ Single instance app
- ✓ No monitoring
- ✓ No database replication
- ✓ Authorization was inside a monolithic app
- ✓ Tightly coupled code
- ✓ Lots of Joins and Updates while working with DB
- ✓ No tech debt culture





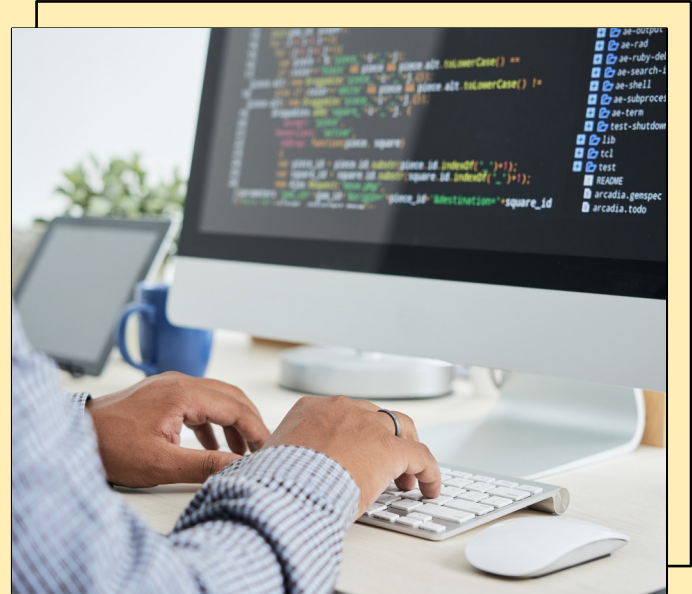
-  System unavailability
-  Low recovery rate
-  Slow TTM
-  Complexity in incident's investigation
-  Difficulties in scaling teamwork
-  Tons of experience and fun



And we need some in-flight changes

ONLINE


- 1 Logs and monitoring
- 2 Scaling
- 3 Code optimization
- 4 Data
- 5 Support
- 6 Technologies



LOGS AND MONITORING



 1 Basic business monitoring

 2 API

 3 Workflow

 4 Clients

SCALING

1

Scale workers
carefully

2

Readiness – simple

Liveness – might be
dangerous

BASIC CODE OPTIMIZATIONS



1 Based on the load profile

3 Number of queries in Database

2 Algorithms and N+1

TO KEEP YOUR DATABASE ALIVE



1

Make a replica set

No time to change the model in the DB....

2

Create additional indexes and denormalize data

3

Remove data smoothly and periodically

SUPPORT



Graceful Shutdown



Reboot the logical nodes of your app



Incident's review

Technologies



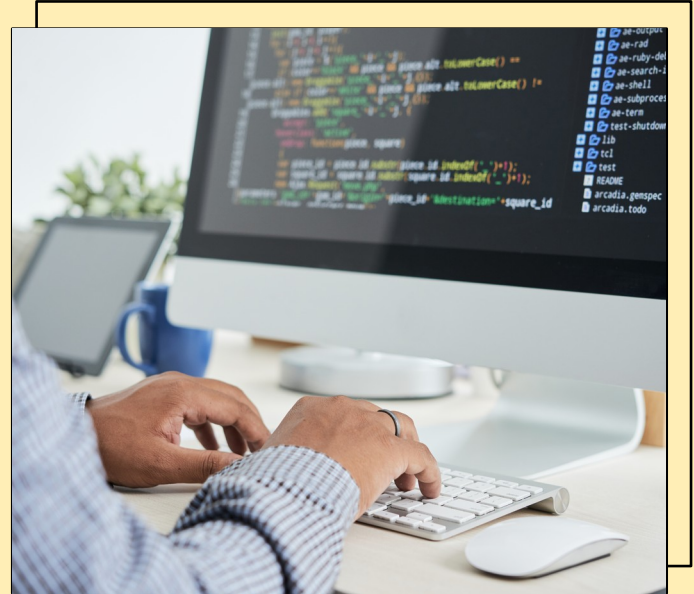
Unnecessary technologies



Sophisticated technologies

ONLINE

- 1 Logs and monitoring
- 2 Scaling
- 3 Code optimization
- 4 Data
- 5 Support
- 6 Technologies



Database info model

Tech debt

Scaling

Architecture

DATA



Data is one of the foundations of your app

Data influences not only the application efficiency but also the whole team efficiency



In the MVP stage
We can design data model based on a business context.
Split data.



LOAD SCALING



Do we need
microservices
in transition
period?



For our case

Distributed
monolithic app
was a saver



On MVP stage

Split your
application
modules

WORKING WITH TECH

DEBT



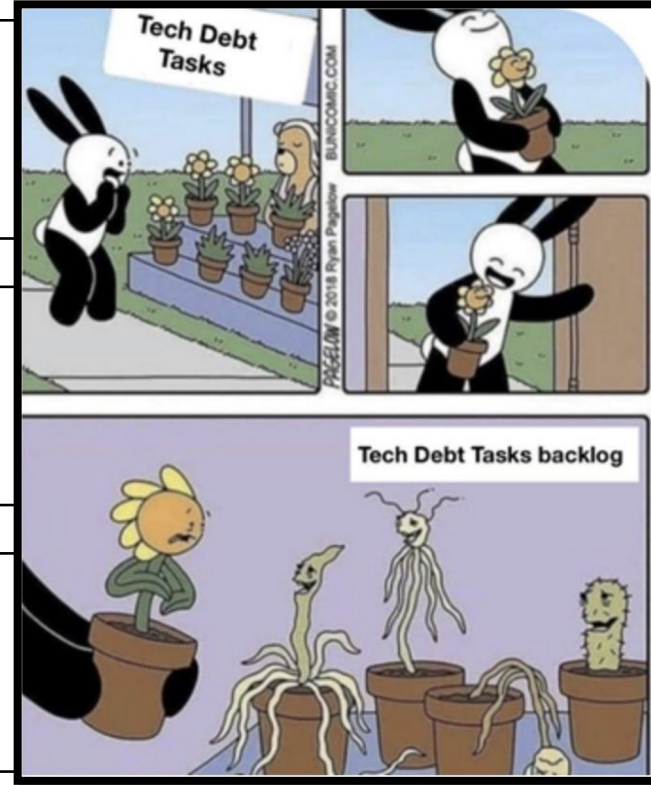
Tech debt is just a string in Excel,
If it doesn't have a business stakeholder



Flexibility is more important than the
“proper” app



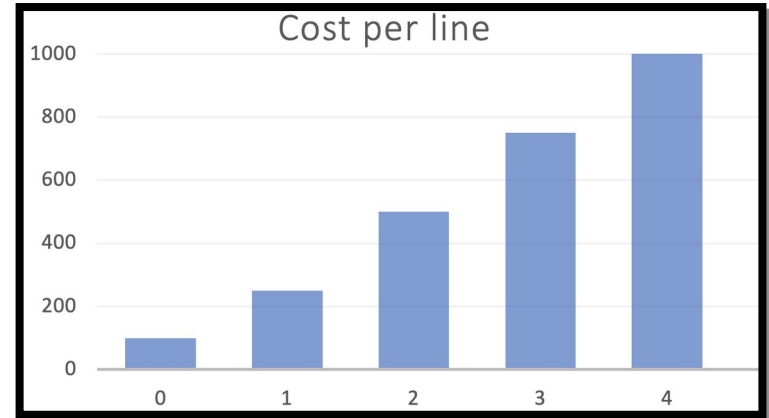
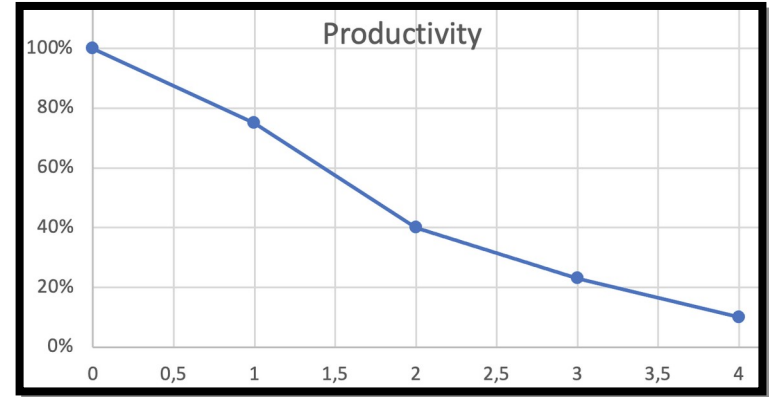
What you could do
Just, please, fix your “not so good” decisions



THE PRICE OF SACRIFICE



Development time
should depend on the
scale of changes, not
on the complexity



ARCHITECTURE



Architecture is about flexibility.

It gives you more options to choose from and more time to think.



Conclusion



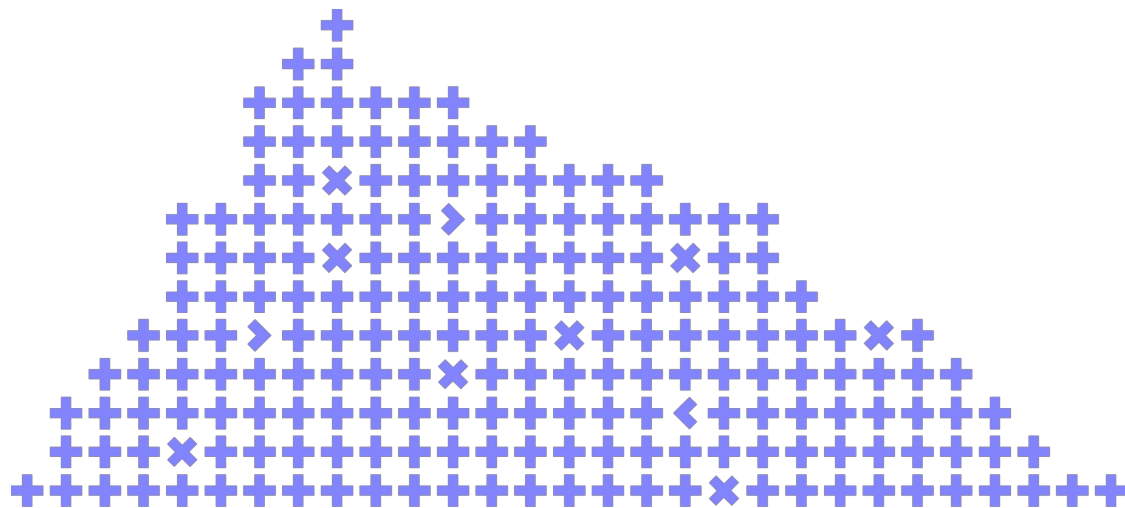
Conclusion



Give your feedback and comments via link

Contacts

Telegram: @av_dashkevich



Co-organizer

Yandex